Fall 12-1-2000

Business Case for Traffic Engineering

Hartzell Dave
Dakota State University

Business Case for Traffic Engineering

Dave Hartzell

Dakota State University

Project Committee Members:

Stephen Krebsbach Committee Member	Minhua Wang Project Technical Advisor	Terry Dennis Student Advisor

Contents

Abstract

This project and the accompanying case studies attempt to investigate and implement new network technologies in the realm of Traffic Engineering.

Traffic Engineering provides a set of tools for network engineers and administrators that allow them to manipulate the network's behavior in a manner that better suites the needs of the users and their applications.

There is a legitimate business case for Traffic Engineering. New applications and uses for the Internet are causing a shift in the type of traffic that was once common. This shift includes new traffic patterns that are generated by services like voice and video, peer-to-peer applications, and real-time data services.

Left alone, the Internet may have difficulty supporting these new applications. Applications currently consider the network 'dumb' or incapable of reliable data transfer. Traffic Engineering techniques permit the owners of the network to make it smarter, by which has the potential to reduce cost, increase performance, and add value to the network.

Traffic Engineering technologies were implemented on real-world next-generation Internet 2 networks owned and operated by midwestern universities for research and education. Traffic Engineering techniques like Quality of Service and Multi Protocol Label Switching were investigated, and if possible implemented, tested and measured on the Great Plains Network and the South Dakota Research and Education Network.

Part I:
The Technologies of Traffic Engineering

Part I:
The Technologies of Traffic Engineering

1. Defining Traffic Engineering

The Internet is playing a major role in our society and economy by linking millions of people, businesses and governments together via a self-policing, loose confederation of computer systems and networks. Currently, all of the activity on the Internet including real-time data flows, voice, video and data transfers all occur in a fashion known as 'Best Effort.'

Best Effort, or BE, means that when a stream of data is sent across a network, the data is sent with the hope that an event such as network congestion will not occur during transit to cause data loss. Short of a catastrophic failure, there no mechanisms in place on the public Internet that will adapt to changing network conditions such as congestion.

Internet Engineering Task Force: Traffic Engineering Working Group

Traffic Engineering may be defined as the adjustment or tuning of the network to optimize performance. The Internet Engineering Task Force (IETF) defines Traffic Engineering as

> *"...that aspect of Internet network engineering concerned with the performance optimization of traffic handling in operational networks, with the main focus of the optimization being minimizing over-utilization of capacity when other capacity is available in the network. Traffic Engineering entails that aspect of network engineering which is concerned with the design, provisioning, and tuning of operational internet networks. It applies business goals, technology and scientific principles to the measurement, modeling, characterization, and control of internet traffic, and the application of such knowledge and techniques to achieve specific service and performance objectives, including the reliable and expeditious movement of traffic through the network, the efficient utilization of network resources, and the planning of network capacity."* [TEWG1]

The results of Traffic Engineering

Data generated by two users exchanging a text-based email will be treated with same priority as the data generated by a real-time voice conversation.

As far as the network is concerned, all data is equal and treated the same, when in reality it may be far more important to deliver real-time data first, a stock transaction second, and the email last. Often, an application like email is not expected to be opened immediately, and this application tolerates latency in delivery. Users have different expectations for the different network applications that they are using. The two computers involved in a transaction will ensure that the email is delivered intact, even if data loss occurs. This is a function of the Transmission Control Protocol, which guarantees reliable delivery of data over the Internet. TCP does not guarantee data delivery within certain time restrictions.

The author's part in this project includes the deployment, testing and measurement of Traffic Engineering technologies on the Great Plains Network and the South Dakota Research and Education Network. New methods and deploying and testing Traffic Engineering Technologies were attempted, with successful results. Part I of this paper is intended as background information that was studied so the best possible outcome could occur while implementing the project. Part II discusses the deployment, testing, and measurement of the Traffic Engineering technologies.

2. Business Case for Traffic Engineering

It can be argued that the Best Effort model significantly contributed to the Internet's success through the development of adaptive protocols like TCP. But we are nearing a point in the life cycle of the Internet in which the desire to transmit and receive multi-service data is emerging. New applications like voice, video and real-time applications are being developed. The Internet can be a *multi-service* network, but traffic engineering techniques need to be deployed before that is becomes a reality. [AVVID]

In order to deliver these multi-service applications over a network, certain steps need to be taken in order to ensure that the applications receive the network characteristics required to perform efficiently. These mechanisms can be Quality of Service algorithms or Multi-protocol Label Switching deployed within the network.

Link Over-Utilization

Wide Area Network (WAN) links are expensive in terms of financial cost. For a single T1 line (1.544 mbit/s) to an Internet Service Provider (ISP), an organization can expect to pay approximately $1000 per month, plus other fees.

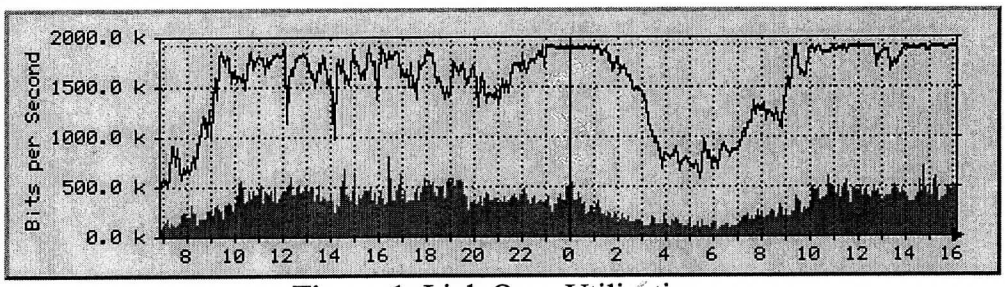


Figure 1: Link Over-Utilization.

If a link reaches a utilization point where users experience unsatisfactory performance for their applications, there may be a lack of provisioning. Figure 1 illustrates link over-utilization [SMD98]. Often an organization will abstain from purchasing additional bandwidth due to the added expense. Quality of Service algorithms can be enabled on existing links to improve application performance, without added cost. Existing network equipment (such as a router) may already have the capability to implement QoS at no additional cost.

Link Under-Utilization

The LAN often has much more bandwidth available due to the lower cost of implementing inexpensive technology like switched Fast Ethernet. WAN links may go 'under-utilized' as far as user experiences and measurement is concerned. Real-time applications may not perform adequately on a link that is considered under-utilized if certain usage conditions occur.

A phenomenon on a network link known as *bursting* can be very disruptive to some network applications. Bursting often appears as a short-lived, high volume peak on a measurement graph. Bursting can effectively utilize all of the available resources forcing other applications to compete for access. Figure 2 illustrates bursting on an Internet link.

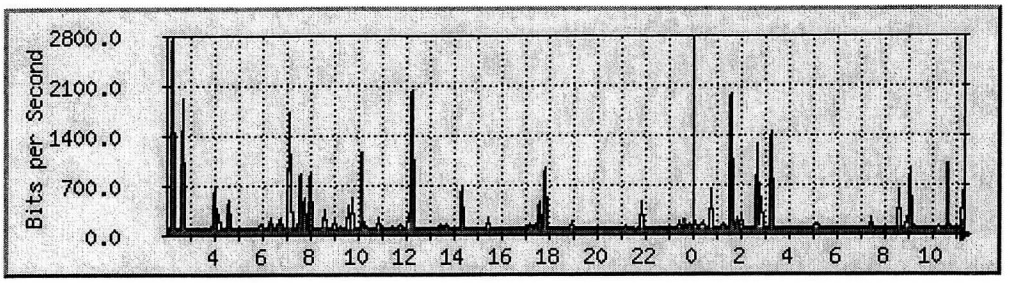


Figure 2: Example of Bursting on an Internet Access Link

In order to reduce the impact of bursting on a link, Quality of Service techniques can be implemented so that other traffic does not have to compete with the bursting traffic. Higher priority would be given to voice traffic, while bulk data traffic would be given a lower priority, since the voice application is much more sensitive to loss than the bulk data transfer application.

Traffic Engineering and QoS can save money by increasing the network's role of intelligently managing bandwidth, and at the same time increasing user application performance.

3. *Multiprotocol Label Switching*

Multi-Protocol Label Switching, or MPLS, is a recently developed traffic engineering technology for use in the WAN infrastructure. MPLS allows engineers to manipulate traffic flows in networks by using methods that easily scale well to large backbone infrastructures.

Over-utilization and under-utilization of network paths within a WAN infrastructure is common. Certain links are highly utilized, and other links are often underutilized. A network provider pays for a link regardless of the utilization, and there may be a desire to manipulate traffic flows to distribute traffic from highly utilized links to underutilized links.

MPLS technology also introduced a new method of performing large scale, high-speed Virtual Private Networks. Before MPLS based VPNs, organizations wanting a private WAN infrastructure had to purchase a second WAN infrastructure for their private network, which brings added expense. With MPLS based VPNs, these services can be integrated into one link using the ISP link for both networks.

MPLS Concepts

Destination based routing attempts to route packets along the shortest path from the source to the destination. While this is typically the most efficient mechanism, there may be a requirement to select an alternate path for specific customers.

An MPLS enabled network merges Layer 2 technology with Layer 3 technology. A *label* is inserted between the layer 2 header and the layer 3 header, as shown in Figure 3. Now, instead of a router making a routing decision based on the destination address in the Layer 3 IP header, the MPLS enabled router makes a forwarding decision based on the label definition. This method is analogous to the Asynchronous Transfer Mode (ATM) Virtual circuit (VC) switching function.

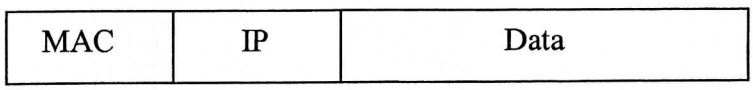


Figure 3a: A Standard IP Packet

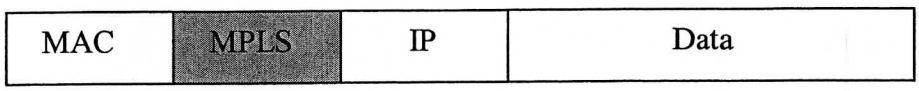


Figure 3b: MPLS Labeled IP Packet

MPLS requires a Forwarding Equivalence Class, or FEC. As a packet enters the MPLS network, it is labeled once according to the FEC. A FEC is distributed among all of the MPLS enabled routing devices within the provider's network. A FEC is defined on a per customer basis, and tells the network which path a certain labeled packet is supposed to

take through the network. Thus, a forwarding decision is made only once in a network instead of multiple times. The rest of the MPLS network now knows how to forward an MPLS labeled packet. The alternative to MPLS is the traditional paradigm where the routers involved with routing the packet make a forwarding decision based on the destination IP address.

To distribute forwarding information within an MPLS enabled network, the Label Distribution Protocol (LDP) was defined. LDP distributes Label and FEC information to all MPLS routers in a network regarding what action should be taken when a labeled packet arrives at a router. Using LDP, an MPLS network builds Label Switched Paths (LSPs), which are essentially end-to-end tunnels.

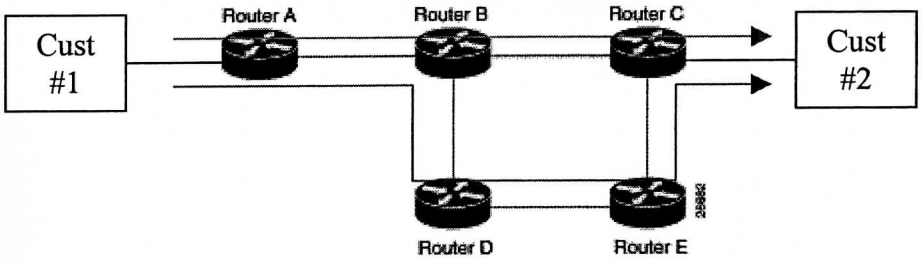


Figure 4: A common infrastructure for a network provider.

MPLS Label Switched Path Application

In a typical IP network, a packet is routed according to the destination, and the packets usually traverse the network using the shortest path route. In Figure 4, Customer #1 connected to Router A wanting to transfer data to Customer #2 would pass through Router B. Unless the link between router B and C is down, Router D and E would not have anything to do with a data transfer between customers connected to Router A and Router C. The blue line represents the shortest path for these customers.

Now assume that the link between Router B and Router C is congested. Performance of applications is being impacted, and customers are experiencing poor application response. With an MPLS enabled network, it is possible to define a new path across the network without having to install new network connections or add routers. It is likely that other customers are also experiencing poor performance due to the congestion.

With MPLS, it is possible to tell the network to send traffic destined for Customer #2 along a different path other than the shortest path route. The red line indicates a new path for only customers #1 and #2. Other customers connected to Routers A and B may still traverse the shortest hop path. By moving some customers and not others, congestion can be alleviated.

As Router A gets a packet from Customer #1, it will put a label onto the packet. Using the label information that has been distributed to all routers in the network, Router A knows to

pass this labeled packet to Router B. Router B sees this label, does a lookup in the FEC database and knows to pass the labeled packet onto Router D.

Router E will receive the MPLS labeled packet and pass it on to Router C, where Router C will discover (by looking in the FEC database stored locally) that it is the end point and egress router for the MPLS Label Switched Path. Router C will now remove the label from the packet and pass it on to Customer #2, as it would a normal IP packet. Customer #2 does not know that the packet was label-switched across the service provider's network.

The Merging of Layer 2 and Layer 3

MPLS merges Layer 2 of the OSI model with Layer 3. Before MPLS, source based routing (where a forwarding decision is made according to the source of the packet, not the destination) was computationally expensive for routers to perform and did not scale beyond one router. With MPLS and LDP deployed in a service provider's network, flow-based routing can now be achieved by telling only one router (the ingress router) what action to take, and what path to take. Without MPLS, source based routing would have to be used, which requires manual configuration of each router in the desired path.

When MPLS technology was first emerging a few years ago, it was common to hear that existing routers are computationally overloaded by having to perform a route lookup for each packet in a database of approximately 100,000 routes. While this may be true, it was also often heard at the same time that MPLS would 'save the Internet' by reducing CPU load on routers, since MPLS does not use the large database of routes.

Since then, new technologies such as faster CPUs and new lookup algorithms have been able to keep up with the ever-growing routing table. At the time of this writing, the current size of the global Internet routing table was approximately 125,000 routes*.

Another setback of MPLS today is that there are no inter-domain MPLS protocols. A service provider can deploy MPLS within its own administratively controlled system, but if an Internet Service Provider is required to pass a packet to a second Internet Service Provider, it must do so using the traditional IP routing methods. While MPLS scales well, there is no defined procedure for MPLS signaling beyond one's own autonomous system.

Regardless of the Inter-domain issues with MPLS, service providers and ISPs are deploying MPLS within their networks to more cost-effectively provide better service to their customers.

---

* For current routing table conditions, see http://www.telstra.net/ops/bgp/

4. Quality of Service

Since users communicating over the network have no control of the network itself, protocols where developed for use by the end-stations that ensured data delivery at the expense of delivery time. This paradigm continues to work well for the data-carrying model of the Internet, but with new multimedia applications, intelligence must be added to the network.

QoS – A Definition

Quality of Service may be defined as an attempt by the network to ensure data delivery within bounds of values concerning delay, jitter, bandwidth or loss. Quality of Service can further be defined as the collective effort of service performance, which determines the degree of satisfaction of a user. [Huston00]

There are several reasons for Quality of Service (QoS) mechanisms to be put into place on today's Internet. Before discussing these reasons, it is appropriate to more adequately define the term 'Quality of Service'.

In it's simplest sense, the word *Quality* in QoS may be defined as *superior performance.* Network performance attributes often coincide with quality, such as bandwidth, loss, delay, and jitter. A reliable, high quality network will often minimize data loss, minimize delay, minimize jitter, and maximize throughput. [Huston00]

With regards to network performance parameters, quality may not necessarily mean minimizing data loss, delay and jitter or maximizing bandwidth utilization or throughput. Quality may also mean restricting or regulating a certain network application to certain bounds. An example of quality in this reference might mean limiting a bandwidth intensive peer-to-peer application. Limits can be put on the amount of bandwidth a user can consume, leaving network resources for the other users.

It is also important that all users have fair access to the network. Some applications (such as Voice traffic) need specific network characteristics to operate adequately, while other users may need to consume more of the available bandwidth. QoS provides algorithms that attempt to ensure equal access to the network resources for all users. One or two monopolistic users could potentially consume all of the network resources, leaving few resources to the other users.

The Best Effort Internet

Until recently, the Internet and IP based data networks that have been implemented have been optimized for data transfer. As time progresses, new applications are being developed and the designers and users want to use one network for all of their communications needs. Applications like voice, video and real-time applications need certain network characteristics in order to operate properly.

Since today's Internet and data networks are Best Effort, they are considered 'fair' in the sense Internet protocols like the Transmission Control Protocol (TCP) attempt to fairly share the network. New voice and video applications may want to share the available network characteristics in a different manner from that of the TCP bulk data transfer model so prevalent on today's Internet.

Voice Traffic

Applications like Voice over IP (VoIP) need special treatment in order to achieve an acceptable quality phone call over the Internet. An application like VoIP needs low loss, low delay and low jitter characteristics over the network, but it does not need large amounts of bandwidth. Using voice compression techniques, a 56 kbps modem link to the Internet may support two to three voice calls over one phone line.

Another characteristic of VoIP is that traffic generated by two end-stations involved in a VoIP communiqué typically generates a Constant Bit Rate, or CBR. This is a classification used to describe certain traffic patterns that do not change sending rates. The graph in Figure 4 illustrates a CBR network flow.

Using Quality of Service algorithms, the network can give the VoIP application the resources it needs, while providing adequate bandwidth for a bulk data transfer application. The network should give low latency, low loss, high-priority treatment to voice traffic if network resources are available.

Video Traffic

Video conferencing is becoming a popular tool in the Internet today, and this trend is expected to grow. New advances in codecs and video technology are providing collaborators with efficient, inexpensive network applications that allow real-time, face-to-face communications over IP based networks. The development of the H.323 protocol defines a set of common video, audio and signaling protocols that allow interoperability.

With an application like video conferencing, it is important to deliver the video stream as quickly as possible across the network with low loss and low latency. This will ensure that the users on the opposite ends of the conference are able to interact as if they were standing in front each other.

Real-Time Data Services

Other applications that may need special access to the network are real-time data applications like stock market transactions and other 'critical' applications in which a certain amount of time to react to a condition is required. A more official definition of real-time is:

> *...a level of computer responsiveness that a user senses as sufficiently immediate*...[whatis1]

Currently, it is very difficult to deliver these service requirements with the Best Effort model of the Internet today.

With other types of near-real time network traffic like voice and video applications, we can handle loss somewhat gracefully. Real-time data applications will require a guarantee of end-to-end delivery within certain time bounds in order to satisfy the definition of real-time. This will be difficult even with QoS mechanisms deployed on the Internet today unless the user's definition of real-time is quite wide and flexible.

Architectures for Quality of Service

Today, there are two architectures for accomplishing QoS on existing IP based data networks. One model is with a Differentiated Services approach, and the other is the Integrated Services model.

Integrated Services, or IntServ, was a model for QoS developed around 1994 and defined by the IETF RFC 1633. The concept of IntServ assumes an active role must be played buy the end stations and the network in order to achieve an acceptable level of service. According to their application needs, the end-stations will signal the network with a request for certain levels of service. If the has sufficient resources, it will reserve and provision network resources in terms of latency, jitter and bandwidth from end-to-end.

The IntServ model assumes that simple packet prioritization is not enough to satisfy the requirements of all of the varying applications that could potentially exist. IntServ is a complicated architecture that may not scale well in the Internet today. Also, IntServ requires network and end station support of the signaling protocol known as the Resource Reservation Protocol, or RSVP.

Differentiated Services, or DiffServ, is a model proposed later by the IETF to achieve some levels of Quality of Service on the Internet. The DiffServ architecture was defined to address the concerns over the scalability and other complexities introduced by IntServ.

The DiffServ paradigm defines a simple packet prioritization scheme along with Per network Hop Behaviors (PHBs). Essentially, it is up to the end-station to define the relative priority of its data that gets sent out to the network. If the end-station's application is a high-priority application, it will mark the packet with a high-end six-bit number known as the DiffServ Code Point, or DSCP. If the end-station's application is generating low-

priority traffic, it will mark the packet with a lower DSCP indicating this packet has a lower priority when compared to that of a higher mark.

Currently, there is more DiffServ research being done throughout IP networks including several of the Next Generation Internet networks like Abilene (Internet 2) and NASA's Research and Engineering Network. Since DiffServ is simpler to support and develop, many vendors are choosing to support this architecture before that of IntServ. Also, since the vendors are leaning toward an initial DiffServ architecture, most networks will probably deploy DiffServ first.

Regardless of the architecture chosen, there are some basic techniques used by both the DiffServ or IntServ like priority queuing, rate limiting, traffic-shaping and policy routing.

Quality of Service Algorithms and Methods

Before discussing these algorithms for performing QoS functions, it is important to define a queue. A queue is a place of temporary waiting. With respect to an Internet router, it would be a memory region where an IP packet would be stored before it gets be transmitted. Queues are used in the event of a non-idle interface, or where the interface is busy doing something else. It is very possible that data will arrive at a router from multiple input interfaces and be destined for one output interface. When this event occurs, the data (in the form of IP packets) will need to be queued for transmission on the outgoing interface, since the interface can only transmit one at a time.

A queuing algorithm known as First In/First Out is commonly used with IP routers today. Essentially, the first IP packet entering a router will be the first packet leaving the router. With respect to a router's queue, the first packet into an interface queue will be the first packet out of the queue.

If an interface queue on a router becomes too full, the router will discard the packet without taking any further action. This is known as Tail-Drop. It is up to the end stations to discover the packet loss and recover gracefully. TCP is not efficient with applications like voice and video conferencing since the end stations do not have to time to detect a packet loss and retransmit. It is better to add intelligence to the router's queuing mechanism so that the voice and video data are not dropped. It would also be advantageous if the router would transmit the voice and video traffic before any other traffic like bulk data transfer traffic or web traffic.

Queuing 101

In order to effectively transmit high priority, low loss, low latency traffic like voice and video within the current paradigm of IP router architecture, multiple queues must be created and assigned to an interface instead of the common single queue per interface.

When data passes through the router, instead of placing the data to be sent on a single queue, the router first examines the packet for distinguishing criteria like a DiffServ Code

Point, source IP address, destination IP address or other protocol information. After matching certain criteria, a voice over IP packet may be placed into an alternate queue, such as a high priority queue.

Assuming there are two queues, a low priority queue and a high priority queue, when the router interface is idle, it will service the higher priority queues first and then send data from the other queue. Figure 5 illustrates the low/high priority queuing mechanism.

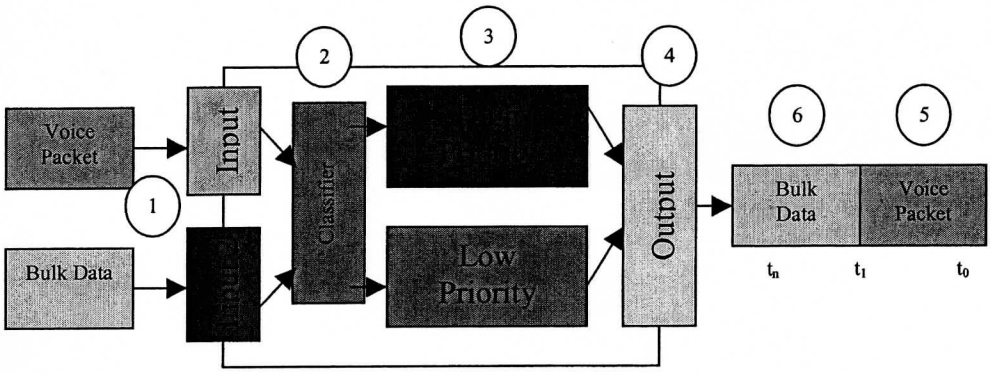


Figure 5: Packet Queuing

The above example is only one method of performing some sort of QoS, and may not be effective if there is a large amount of voice traffic, since the router will be serving the high-priority queue constantly.

Queuing Methods

In addition to the high priority queuing method previously discussed, other algorithms have been developed to provide alternatives to those wishing to implement QoS. Several algorithms have been developed, and network engineers and administrators are expected to evaluative their needs and deploy the best algorithm to their situation.

Some of the available queuing algorithms today are:

- *Low Latency Queuing* – Low Latency Queuing or LLQ, is very similar to the example described earlier where there are two queues, and the high-priority queue is serviced before any other queues. This algorithm can be related to the First Class line at an airport check-in.

- *Deficit Round Robin* – Deficit Round Robin, or DRR, is a queuing method in which several queues are defined and the router places differentiated packets into the appropriate queue. Then, when the router interface is ready to transmit data, it services each queue in a round-robin fashion. A deficit counter is kept that tracks how many bytes each queue has transmitted each round. The deficit counter is configurable to meet the needs of the network administrators. DRR is

similar to an entrance to a roller-coaster ride where several lines are formed, and each pass of the ride allows a certain number of people to ride from each queue.

- *Weighted Fair Queuing* – Weighted Fair Queuing, or WFQ, is an attempt by the router to classify packets by service characteristics and then sort the packets with common characteristics into similar queues. For example, all FTP would be put in one queue, and all Telnet into another. Weighting would be given to certain queues indicating a priority. For example, an implementation of WFQ might automatically give Telnet a higher priority than FTP. Recent implementations of WFQ can be tuned by the network administrator to give a higher priority of a certain type of traffic, like VoIP.

Congestion Management

All of the queuing algorithms discussed above fit into a category of queue management techniques called *congestion management, or CM.* CM is an attempt by the network to handle a congested link or interface with queuing strategies like WFQ, DRR, or LLQ. These solutions attempt to answer the question, *'How can we provide a high quality of service to a network while congestion is occurring?'*

Congestion Avoidance

Sometimes it is better to attempt to avoid congestion completely. This can be done using some congestion avoidance algorithms on the network, and there are three prominent algorithms for congestion avoidance, *policing shaping* and *detection.*

Traffic Policing, also known as rate limiting, is a method of eliminating or marking network packets if their rate on the network exceeds a pre-defined threshold. For ISPs, policing means that they can throw away a customer's traffic if it exceeds what the customer has contracted for.

Policing also allows for an ISP to mark customer traffic as 'discard-eligible' meaning that a flow of traffic has violated a policy somewhere in the network. Later, if necessary, another part of the network can discard the traffic should congestion occur.

Congestion avoidance mechanisms are more applicable to the Integrated Services architecture. For example, should a network user signal the network for a bandwidth reservation, the network expects the end users not to exceed their reservation amount. If the end exceeds the reserved amount of bandwidth, the network will drop that traffic.

Another congestion avoidance mechanism similar to policing and rate limiting is *traffic shaping.* Traffic shaping is a function where traffic is buffered and 'smoothed' by the network within a defined limit. This is similar in concept to policing, except those packets are not discarded if they exceed their contracted limit.

A shaper delays excess traffic using a queuing mechanism to hold packets and regulate the flow when the data rate of the end source is exceeding the contracted amount. Traffic shaping is gentler with respect to the end-to-end flow of packets. Instead of the end stations performing data-loss discovery and packet retransmissions, there is a steady flow of packets with typically no loss. The end-stations adapt to the allowed limit gracefully through the flow control mechanisms of TCP.

Traffic shaping does not come without cost, since the network is required to buffer more data and regulate data flows on a per-instance, per-flow basis. This not only consumes more memory, but also consumes CPU cycles of network routers and switches, which can impact overall performance of the network for all users.

The last congestion avoidance mechanism is Random Early Detect, or RED. RED is a queuing mechanism that attempts to handle interface congestion before it occurs. RED also does not attempt to re-order the delivery of packets, and uses the FIFO queuing algorithm by default. The RED algorithm monitors the average depth of an interface's queue. If the queue reaches a high-water mark, say 75% full for example, then the RED algorithm will randomly select and discard a packet that packet. See Figure 6 for an example of the RED drop probability algorithm. As the average queue size increases beyond the minimum threshold, the probability of a packet being dropped increases. If RED were not enabled on an interface, the drop probability would be zero until the maximum threshold were reached, then the probability would immediately become one.

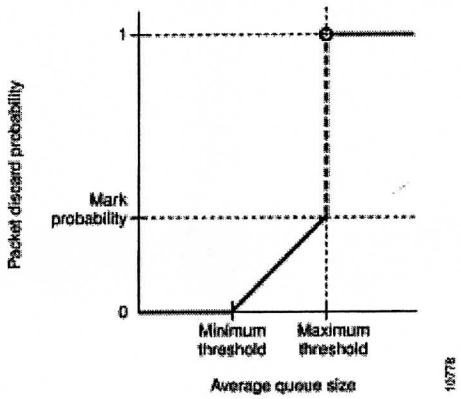
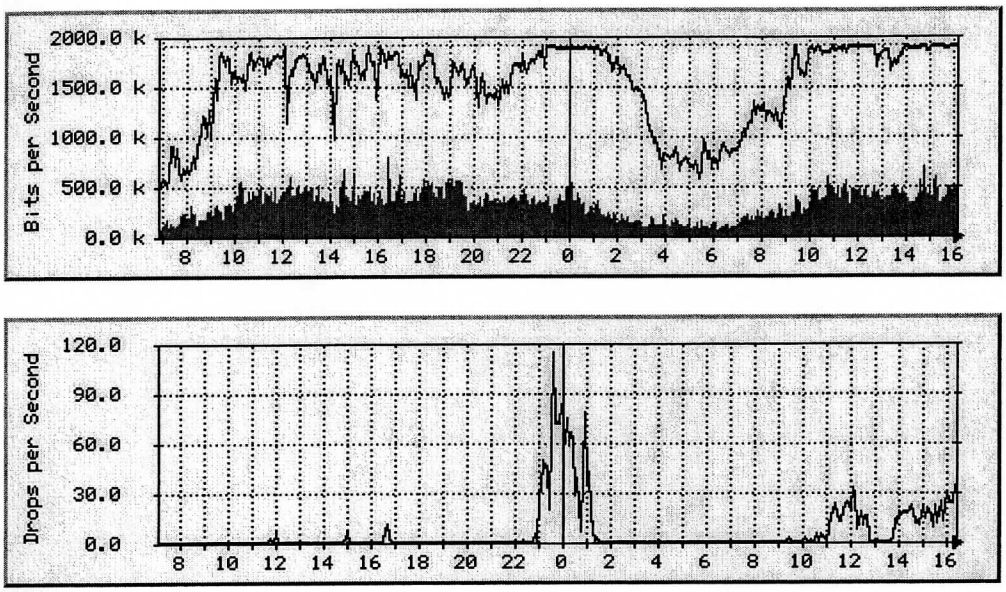


Figure 6 – RED drop probability.

The intent of dropping packets will signal to the end stations communicating that there is congestion in the network and that they need to reduce their transmission speeds. This is handled automatically by TCP on the end hosts. RED is used to reduce the effects of global synchronization, where multiple end hosts communicating over the same link experience congestion at the same time. All of the end stations then suddenly reduce their transmission speeds at the same time. This continues to occur in an oscillating fashion. See [FloydVanRED93] for more details.

Figure 7 clearly shows the effects of implementing RED on a highly utilized interface and link. RED was turned on at approximately 1000 hours. Notice how the line was used more

efficiently to its maximum capacity of 2000 kbit/s and how the interface drops become smoother and more frequent between 1000 hours and 1600 hours. [SMD98]



Figures 7a and 7b– Effects of RED implementation.

One of the original authors of RED has suggested an alternative to dropping packets randomly when the average high-water mark has been exceeded. Instead of dropping the packets, a bit would be toggled by the network in the IP header of the packets that would indicate to the end host that there is congestion somewhere in the network. This is known as Explicit Congestion Notification (ECN) [floydECN]. If a host is ECN capable, and it receives a packet marked by the network as congested, it will reduce its transmission rate. This may be more efficient than requiring the end hosts to discover packet loss through time-out algorithms. ECN is currently in the development stage, but there are some host implementations of this technology. No know router implementations exist, except for Nortel Network's Open IP 2.1 implementation.

When should QoS be used, and is it needed?

It is a common debate whether QoS functions like priority queuing should be used only if the interface is congested or highly utilized. Often though, an underutilized interface sees periods of bursting or high-utilization that may make applications like VoIP video conferencing suffer in terms of perceived performance.

It can also be argued that QoS may be able to deliver superior performance to certain real-time applications that require the best, fastest delivery time possible. Small real-time packets will experience delay when stuck in a queue behind larger packets. But, if the smaller, real-time packets are placed in a higher priority queue than the larger packets, the interface will service those smaller packets first even if the interface is not considered

congested. If the goal is to deliver packets to the end destination as quickly as possible, then QoS may help achieve this goal.

On the other hand, queuing algorithms might increase the delay of *all* packets, since a network device like a router or switch will have to analyze and classify all packets into different queues. The process of analyzing these packets requires more time and CPU cycles thus causing an increased latency for all packets passing through the network device. Whether or not this increased latency impacts perceived performance is another question.

By looking at a typical packet size histogram (see Table 1), we can see if it is desired to send a small 200 byte, high-priority, real-time packet, there is a chance that the small packet may be placed into a queue behind a larger packet. The larger packets will take longer to transmit, thus increasing the amount of time the smaller packet has to wait in the queue.

1-32	64	96	128	160	192	224	256	288	320	352	384	416	448	480	512	544	576	1024	1536
.001	.679	.201	.022	.017	.010	.007	.005	.012	.003	.002	.003	.002	.002	.003	.002	.001	.002	.011	.014

Table 1: IP packet size distribution

Since there are various sizes of large packets, it will be difficult to predict an actual delivery time without using a queuing algorithm like DRR. [shcreedharDRR]

Part II:
Project Implementation

5. *Multiprotocol Label Switching and the Great Plains Network*

The Great Plains Network

The Great Plains Network GigaPoP (Gigabit network Point of Presence) is a high-speed Internet 2 research and education network deployed in Minnesota, North Dakota, South Dakota, Nebraska, Kansas, Missouri, Arkansas and Oklahoma. This network currently utilizes layer 2 Asynchronous Transfer Mode (ATM) technologies along with layer 3 Internet Protocol (IP) routing to manage provisioning and traffic engineering tasks. ATM and IP are successfully operating the network at DS-3 (45 Mbps) speeds, but since the initial Great Plains Network has been deployed, new networking technologies have emerged that can enhance the quality of network service for the peering institutions and their network applications.

The Great Plains Network (GPN) connects the university and state level networks together via a high-speed, next generation infrastructure. The connecting network also provides the state and university networks with access to the Abilene network, which permits access to other gigaPoPs, universities, federal networks and international research, education, and engineering networks.

Part of the original charter of the Great Plains Network was to implement new networking technologies as they become available to enable researchers and engineers to better understand how to build and use high-performance, scaleable Internet infrastructures.

MPLS traffic engineering technology was evaluated for use on the GPN, but due to the infrastructure of the network, deployment of MPLS was not feasible. The backbone of the GPN was insufficient in terms of alternative links and MPLS Routers in order to perform such functions as traffic engineering. There was also insufficient demand for MPLS based VPNs between the universities and various state networks.

It was determined that instead of attempting to deploy MPLS in a network that could not support the technology or in a network where there was no need for the technology, other advanced networking concepts should be investigated. With the recent impact and congesting caused by peer-to-peer Internet applications, Quality of Service seemed a more appropriate study, and remained within the realm of similar traffic engineering technologies.

6. *Quality of Service and the Great Plains Network*

After an investigation of MPLS and the GPN, it was determined that other traffic engineering technologies would be more appropriate and useful toward enhancing network performance for the GPN member institutions.

Nearly every institution and state university on GPN felt the impact of new peer-to-peer applications like Napster and Gnutella. These applications where not intended to function like the traditional client-server methods, but function at the edge of the network, between user PCs, sharing MP3 files and performing distributed computing calculations for projects like SETI@Home and Distributed.net.

Most universities saw their once lightly loaded DS-3 to the GPN network now fully saturated. The peer-to-peer applications were consuming all of the available network resources, forcing more legitimate network applications like research to contend for network resources. Plus, there has been a push to offer distance education over the network using new video conferencing applications.

As a result of the sudden overwhelming network use and congestion, new solutions were needed to cope with the loss of performance that raw bandwidth once provided. Most universities could not afford to add more bandwidth to their WAN connections, but at the same time they needed the network for more appropriate research and education activities. The author of this paper took an action to establish a new program called the GPN Advanced Network Testing program.

GPN Advanced Network Testing Program

The initial goals of the GPN Advanced Network Testing (GANT) are to provide a low-loss, low-latency, low-jitter 'virtual leased-line' (VLL) service. This traffic, when treated properly by the GPN and campus routers, will achieve expedited forwarding (EF) within the bounds as defined by the service level agreement (SLA) made between the GPN and the campus institution.

Further GANT steps could include other QoS and flow-control techniques such as Random Early Detection, Weighted Fair Queuing, and Priority Queuing, for example.

Anticipated uses for GANT

One of the goals of the GANT test program is to establish a low-loss, low-latency, low-jitter premium service for GPN connectors for certain applications. Certain applications like video over IP often demand low-latency and low-jitter network connectivity between end users, but not a large amount of bandwidth. Other applications such as voice-over-IP (VoIP) require a low-loss, low-jitter network service and a very minimal amount of bandwidth. Typical applications like file transfer and web access are not sensitive to delay and latency, and can recover gracefully from lost data, with a complete end product.

The GANT program will provide a premium network service to applications like VoIP and IP or IP multicast video delivery. This service is not intended to allocate large amounts of high-priority bandwidth (> 5 Mbit/s) to connectors, nor is the intent to treat other connector's network traffic with lower priority. Other QoS techniques, such as Random Early Detection or Weighted Round Robin Queuing can be used to increase the effectiveness of a circuit connection and certainly fall within the scope of this test program, should GPN participants be willing to participate.

GANT Quality of Service Technologies

Several methods of congestion management and congestion avoidance are available to the members of the Great Plains Network, with techniques already included in software and routing packages.

Versions of Cisco IOS 12.0 and above contain a rich feature set of congestion avoidance and congestion management tools that can be used to increase the quality of service for connectors and their applications. These methods will be employed on both GPN network equipment and on the connector network equipment. Also, to support end-to-end QoS, individual LANs and workstations should be required to implement a consistent end-to-end SLA.

Examples of congestion management and congestion avoidance available (but not limited to) to the GANT program are:

- Random Early Detection (RED) and Weighted RED
- Priority Queuing
- Custom Queuing
- Weighted Fair Queuing (WFQ) and Weighted WFQ
- Deficit Round Robin (DRR) and Modified DRR (Currently Available on GSRs only)
- Generic Traffic Shaping (GTS, Smoothing)
- Committed Access Rate (CAR, Policing and marking)

Other server and workstation operating systems (like Linux and Windows 2000) contain tools to aid in the deployment of end-to-end SLAs. These technologies include:

- Class of Service (CoS) packet marking
- 802.1Q (Ethernet packet marking)
- DSCP and Type of Service (ToS) bit marking capabilities

Most likely, in order for a successful end-to-end SLA to be established, a combination of these tools will be required. For example, current IP/Ethernet video conferencing systems contain the ability to use Type of Service bit marking. After an end device marks an IP packet with a certain ToS code point, the network will prioritize this type of traffic for expedited forwarding using Custom or Priority Queuing, or MDRR. The network will also ensure that the end device does not exceed the contracted rate-limit using CAR. If this

does occur, the end device over exceeding it's contracted limit could have its packets dropped by the network, resulting in poor performance of the application, with little degradation to the network's resources.

Monitoring the use of GANT

Using software counters on the network core and edges, connectors and staff will be able to monitor the status of their traffic as it applies to the expedited or conditioned traffic. Management information on the network equipment can provide some detail as to how traffic is being treated.

Cisco Systems has recently released a software package available at no cost for specific hardware platforms that aids in the creation, deployment, and monitoring of QoS policies and features. This package, Quality of Service Device Manager 2.0 (or QDM 2.0) can be installed on a router hardware platform and allow for remote monitoring and administration via a standard, Java compliant web browser.

QDM 2.0 includes tools such as a QoS Wizard and real-time graphing of traffic activity or QoS activity. It is intended that the GANT program will use QDM 2.0 for monitoring and for deployment of QoS policies.

GANT Connector Requirements

GPN connectors wishing to participate in the GANT program will be required to submit a formal request to the mailing list gp-qos@greatplains.net. The GPN Engineering staff will review this request, and all reasonable requests will be evaluated for feasibility, and upon approval, all requests will be acted upon as soon as possible.

The GANT program plans to inter-operate with the Abilene Premium Service program, and the GPN Engineering staff will be required to contact the Abilene NOC to coordinate an equivalent request on the Abilene network to ensure an end-to-end SLA.

Connectors will be required to ensure that their campus and local area networks provide equivalent SLAs within their administrative domain. Connectors will also be responsible for classifying their traffic so the network can appropriate treat their traffic to match that of the SLA.

GANT Today

To date, there has been very little deployment of GANT technologies other than in the backbone of the network. The GPN owns two Cisco 12000 Gigabit Switch Routers (GSR) that connect the network to Abilene in two different locations, one in Kansas City and the other in Minneapolis. These routers also provide transit between the state networks, as shown in Figure 8.

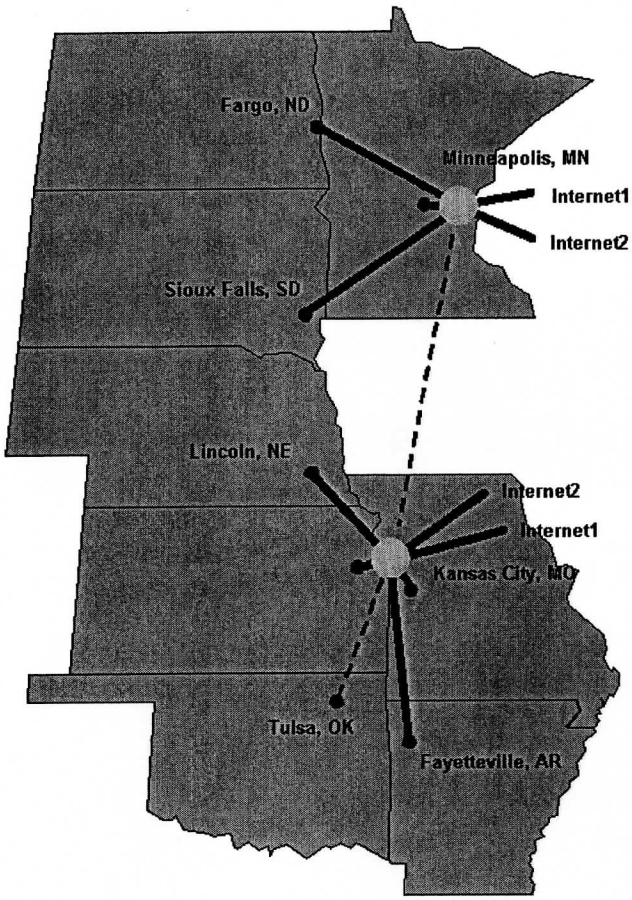


Figure 8: The Great Plains Network.

The dotted line actually represents the Abilene network. Due to funding cuts, the GPN uses the Abilene Internet network to link the Minneapolis and Kansas City nodes. Also, traffic destined for other nation-wide universities, government institutions, and international network peers leaves via the two Cisco GSRs. These two GSRs each have an OC-12 Packet Over SONET (POS) interface to an Abilene Router, one in Indianapolis and one in Kansas City. These interfaces now have QoS functions enabled on them and are discussed in Section 7 of this paper.

The first step in implementing the GANT was defining the scope of the project. The second step was enabling the core of the GPN with the advanced services needed to increase performance. The third step will be assisting GPN member institutions by deploying advanced services in their networks. As new Internet applications develop, there is an expected rise in the amount of GANT deployment within the GPN.

7. GPN Premium and Scavenger Service

There have been several Internet 2 initiatives to implement value added advanced network services. Most of these initiatives are sourced from the Internet 2 working group known as the QBone working group. QBone is short for QoS Backbone, and is an attempt to define QoS parameters for the Internet 2 and the next generation Internet.

Abilene: The Internet 2 Research and Education Network

The Abilene backbone network provides high-performance best-effort nationwide connectivity to Internet 2 universities and other institutions. Internally, Abilene is a pure packet-over SONET (POS) network, providing coast-to-coast OC-48 (2.4 Gbps) IP transit. Connectors (like the GPN) attach to one of ten Points of Presence (PoPs) or IP-over-ATM access circuits, running at OC3, OC12, or OC48 speeds.

Abilene Premium Service

Abilene is currently deploying QoS techniques on the backbone and ingress/egress points of the network as part of the Internet 2 QBone Premium Service effort. The test program is described in [AbQoS1] and will be deployed in several phases and utilized EF per-hop behavior to implement a service model similar to a VLL.

It is the intent of the GANT program to integrate with the APS program, and to provide seamless transit of EF traffic between all connectors and backbone transit providers. Applications like voice and video are expected to take advantage of the APS program to ensure priority delivery of time sensitive traffic across the network.

Abilene Scavenger Service

Another program just recently defined[ϕ] and in pre-deployment stages is the Qbone Scavenger Service, or QBSS[Ψ]. Scavenger Service is defined as

> *"...an additional class of best-effort service. A small amount of network capacity is allocated (in a non-rigid way) for this service; when the default best-effort capacity is underutilized, QBSS can expand to consume unused capacity. Applications that are relatively tolerant of greater loss, delay, and jitter may mark their traffic for QBSS and receive a level of service that is potentially degraded compared to the default best-effort service."* [QBSSdfn]

QBSS can be applied in many situations. One potential application of the QBSS would be to use QBSS service for long lived, high-volume bulk data transfers. Many researchers and

---

[ϕ] The author of this paper participated in the design team. See http://qbone.internet2.edu/qbss/ for more information.

[Ψ] The Abilene Scavenger Service has been renamed to the QBone Scavenger Service, or QBSS.

science projects often have terra-bytes of information they wish to transfer from university to university. Sometimes this data is shipped via magnetic storage and courier. Some researchers are reluctant to send their data via Abilene because of the potential impact it would have to network performance for the other users. With QBSS, these researchers could mark their traffic as less-than-best effort and perform their transfers without impacting performance of other applications on Abilene.

A second conceptual use for QBSS is simply to resell the unused bandwidth. With this concept, it may be possible in the future for an Internet 2 university to purchase commercial Internet service over the Abilene network. This traffic could be marked as less-than-best-effort so it would not impact performance of the next-generation Internet services.

With most research projects like Abilene, there are technology transfer benefits to the 'real-world', and the QBSS project is no different. With a concept like QBSS, an ISP could sell less expensive ISP service to a business for a reduced cost, as long as their traffic is considered less-than-best effort. Also, it might be possible for an ISP to charge a reduced rate to a customer if the customer marks their traffic as less-than-best-effort.

There are a few unexpected applications of the QBSS program. For instance, universities are marking all traffic to and from the student dormitories as less-than-best effort. Peer-to-peer applications like Napster usually originate and terminate within the dorms, and university Internet 2 links have encountered congestion as a result. With QBSS, if there is bandwidth available, the students can consume all of the available bandwidth, unless non-QBSS marked traffic wishes to use the network.

Because of the fact that the Abilene network is mostly underutilized, limiting algorithms for QBSS rarely activate, but the routers are still placing and servicing the QBSS and best effort traffic in separate queues. It has been suggested that it may be possible for QBSS marked traffic to receive better service when queues are empty and interfaces are not congested, since QBSS is often guaranteed 1% of the available bandwidth![*]

In order for traffic to receive preferential treatment or less-than-best-effort treatment on a network, the IP packets must be marked appropriately by the end hosts or by the first hop router in the network. Packets marked with DSCP 8 are considered less-than-best effort and packets marked with DSCP 46 are considered priority traffic. The six-bit DSCP marking within the IP Type-of-Service byte is compatible with the three bit IP Precedence mapping also within IP Type-of-Service byte.

---

[*] See the discussion between two QBSS Design Team members at http://archives.internet2.edu/guest/archives/i2ss-dt/log200104/msg00004.html .

Below are the exact router commands used to configure the GPN Cisco 12000 gigabit switch routers for Premium Service and Scavenger Service.

```
cos-queue-group pos_tx
 precedence 0 random-detect-label 0
 precedence 1 queue 1
 precedence 1 random-detect-label 0
 precedence 2 random-detect-label 0
 precedence 3 random-detect-label 0
 precedence 4 random-detect-label 0
 precedence 5 queue low-latency
 precedence 5 random-detect-label 0
 precedence 6 random-detect-label 0
 precedence 7 random-detect-label 0
 random-detect-label 0 155 5167 1
 queue 0 99
 queue 1 1
 queue low-latency strict-priority
```

These commands perform several functions. First, the *cos-queue-group* command defines a new set of QoS policies called *pos_tx*.

The *precedence* commands 0-7 define eight queues, which is the maximum amount of queues that the Cisco GSR currently supports with this version of software and hardware.* The number following the *precedence* command not only identifies the queue number, but it also maps directly to the Precedence bits located within the IP header. The IP Precedence bits are compatible with the DSCP methodology, and are set by the end stations' applications or by other ingress network routers.

All queues are using RED for congestion avoidance, and queues 1 and 5 are performing the Premium and Scavenger Service, respectively. Queue 5 is defined as the low-latency, high-priority queue, while queue 1 receives only one percent of the bandwidth, as indicated by the second-to-last line. All other queues receive 99 percent of the available bandwidth.

It may be required that policing be configured in conjunction with DRR queuing in order to keep the high-priority queue from flooding. By not installing policing on a network, there is potential for a situation to arise where a router or network experiences a Denial of Service attack. Attackers could flood the high-priority queue with worthless information, forcing the router to service the high-priority queue constantly. If strict queuing is used (a function of Modified DRR), all other interfaces could be starved, causing no other traffic to pass through a router.

* The current software revision is IOS 12.0(17) S, and the OC-12 POS line-card is the Engine 0 hardware revison.

To configure CAR, the following commands must be configured on the POS Interface to the Abilene network:

```
interface POS0/0
 description Local fiber interface to Abilene
 ip address 164.113.238.194 255.255.255.252
 rate-limit input dscp 46 8000000 1000000 2000000 conform-action transmit exceed-action transmit
 rate-limit input dscp 8 8000000 1000000 2000000 conform-action transmit exceed-action transmit
 rate-limit output dscp 46 8000000 1000000 2000000 conform-action transmit exceed-action transmit
 rate-limit output dscp 8 8000000 1000000 2000000 conform-action transmit exceed-action transmit
 tx-cos pos_tx
```

CAR is configured with the *rate-limit* command. CAR is installed in an outgoing and incoming direction on the POS interface.

These commands match the DSCP values 8 and 46. Since the nature of CAR is to typically drop traffic or re-mark traffic for dropping later in the network, a threshold level must be defined. Typically, if the amount of traffic exceeds the threshold, packets are discarded. If the traffic does not exceed the threshold set, then packets are permitted.

Instead of dropping the packets in this case, all traffic is transmitted regardless of the threshold level of the matching traffic. The rate-threshold is set to 8000000 bits per second, with a 1000000 bit average burst rate, with a 200000 bit max burst rate. If the amount of matching traffic is less than the threshold (conform-action) then the data is transmitted. If the 8000000 bits per second threshold is exceeded (exceed-action) the data is also transmitted. Counter values are kept for conforming packets and exceeding packets, which reflect the amount of DSCP marked data that passes in and out of the POS interface to Abilene.

CAR Statistics can be seen using the 'show interface' command:

```
ks-2.r#sh int rate-limit
POS0/0 Local fiber interface to Abilene
  Input
    matches: dscp 46
      params:  8000000 bps, 1000000 limit, 2000000 extended limit
      conformed 193010 packets, 255822522 bytes; action: transmit
      exceeded 298405 packets, 428993500 bytes; action: transmit
      last packet: 596088ms ago, current burst: 0 bytes
      last cleared 4d02h ago, conformed 5000 bps, exceeded 9000 bps
    matches: dscp 8
      params:  8000000 bps, 1000000 limit, 2000000 extended limit
      conformed 3559363 packets, 3381M bytes; action: transmit
      exceeded 701619 packets, 993655045 bytes; action: transmit
      last packet: 212ms ago, current burst: 0 bytes
      last cleared 4d02h ago, conformed 76000 bps, exceeded 22000 bps
  Output
    matches: dscp 46
      params:  8000000 bps, 1000000 limit, 2000000 extended limit
      conformed 28185 packets, 1768180 bytes; action: transmit
      exceeded 0 packets, 0 bytes; action: transmit
      last packet: 14276ms ago, current burst: 0 bytes
      last cleared 4d02h ago, conformed 0 bps, exceeded 0 bps
    matches: dscp 8
      params:  8000000 bps, 1000000 limit, 2000000 extended limit
      conformed 13799712 packets, 7603M bytes; action: transmit
      exceeded 0 packets, 0 bytes; action: transmit
      last packet: 192ms ago, current burst: 0 bytes
      last cleared 4d02h ago, conformed 171000 bps, exceeded 0 bps
ks-2.r#
```

Cisco provides a private MIB called CISCO-CAR-MIB. This MIB lies under Cisco's private MIB at OID location 1.3.6.1.4.1.9.9.113.

Each application of a rate-limit to different interfaces produces multiple instances within the SNMP table. The two MIB entries that we are interested in are:

```
"ccarStatSwitchedBytes"          "1.3.6.1.4.1.9.9.113.1.2.1.1.2"
"ccarStatFilteredBytes"          "1.3.6.1.4.1.9.9.113.1.2.1.1.4"
```

These values are polled remotely via SNMP and stored or graphed by an SNMP manager. Figure 9 is a graph generated with RRD Tool and SNMP Tools. Appendix A contains experts of software code written to generate graphs and store the SNMP data. The graph accurately reflects the amount of DSCP marked traffic passing in and out of the Kansas City Great Plains Network Cisco 12000. Using CAR it is only possible to monitor occurrences of DiffServ marked traffic, not the actual queuing functions of the router. As of version 12.0(17)S of Cisco IOS for the GSR, there was no way to monitor actual queuing functions. It is an on-going goal of this project to monitor these functions, and add functionality later if needed.

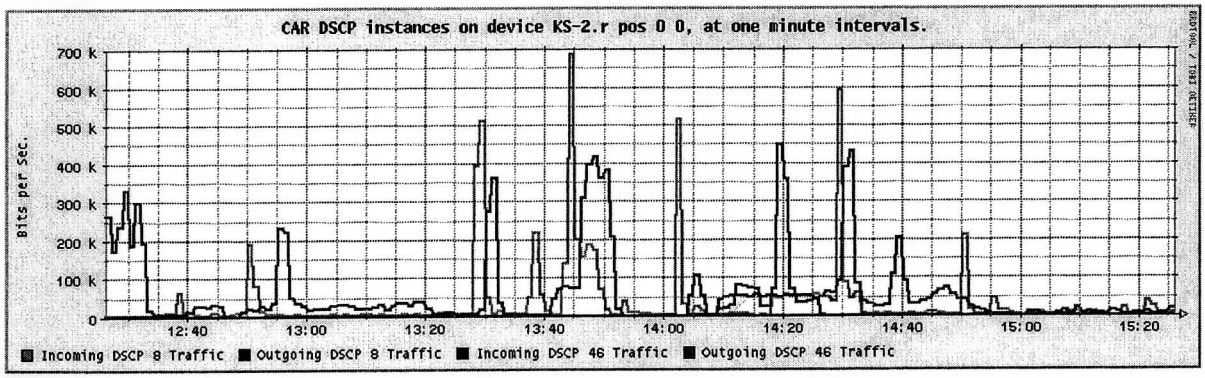


Figure 9- CAR DSCP occurrences on KS-2.r POS

8. Quality of Service and the South Dakota Research and Education Network

Recent funding cuts within Great Plains Network and the South Dakota Research and Education Network (SDREN) have forced the networks to re-home their link to the Abilene Internet 2 network.

Initially, each of the university routers connected via a DS-3 to a GPN GSR in Minneapolis and in Kansas City. Without funding from the National Science Foundation, a cheaper, lower bandwidth method of getting Abilene access was required.

This method required installation of a Cisco 7500 series router at the EROS Data Center, and then bringing in a partial DS-3 from each school to EROS. EROS would have one DS-3 to a GPN GSR in Minneapolis, which could pass traffic to Abilene via the original OC-12 Packet over SONET interface. Each school now has approximately half of a DS-3 connection (20 megabits/sec), but still has high-speed access to Abilene, EROS, and the other two schools.

Distance education classes were being offered between the three South Dakota schools via H.323 video appliances, and with lower bit rates available on the connections, QoS was needed on each campus router and the Cisco 7500 at EROS. This would ensure that the video would not degrade if some other network activity were to occur simultaneously. QoS functions were already enabled on the GSR in Minneapolis.

The Cisco 7500 series router has more QoS functionality than the Cisco GSR routers, but in general is a smaller chassis router supported less bandwidth. The Cisco 7500 is a legacy 'shared architecture' router verses the newer switched back-plane architecture of the Cisco GSR. Essentially the Cisco 7500 is more like a UNIX workstation with highly optimized hardware and software for routing packets, while the GSR's architecture more closely resembles an Ethernet switch.

After several weeks of evaluating the options for QoS, Cisco IOS version 12.1E was selected and installed on the 7500*. Version 12.1E is an experimental train of IOS 12.1 mainline and supports new QoS features that will (if successful) be integrated into the mainline release of IOS.

---

* Actually, this process took about a month and a half with support from Cisco. Version 12.1T was initially selected and installed on the 7500, but failed to work in its entirety and caused the router to become unstable. Special thanks to Stas Shalunov of Internet 2 and George Uhl of NASA Goddard.

It was decided to turn on several QoS features for the universities to ensure a high quality of service on their limited bandwidth connections. Below is a synopsis of the features that are currently installed and operating.*

- Random Early Detection
- Priority service with up to 30% of the available bandwidth
- Less-than-best effort service with a minimal guarantee of 1% of the available bandwidth
- Use of Cisco IOS class-based, command-line QoS definition tools
- DiffServ and IP Precedence matching capabilities

It was also important to turn on similar features at the campus WAN border. Since each campus uses a Cisco 7500 series router, it is possible to fulfill an end-to-end performance requirement that attempts to guarantee a high quality of service for the network users.

Below is partial copy of the QoS configuration used on the Cisco 7500 at EROS. This is a much different set of instructions when compared to the Cisco GSR, since the features supported on the GSR are completely different.

```
class-map match-any qbss
  description Qbone Scavenger Service
  match ip dscp 8
  match access-group name news
class-map match-any Premium
  description QBone Premium Service
  match ip dscp 46
  match ip precedence 5
 class-map match-any default
  match any
```

This groups of commands are *class-map* statements that tell the router what type of traffic to examine. The first class-map command defines the less-than-best-effort service or Scavenger Service, and the Premium matching DSCP 46 or IP Precedence 5 for priority traffic. The last statement is the *default* statement and matches all other traffic that is not marked priority or less-than-best-effort. Also, the QBSS class-map statement matches any traffic that is USENET news. A USENET news feed can easily overtake a 20-megabit per second network connection, so it was decided to put USENET news in the less-than-best effort category.

* It is important to note that the 7500 and 12000 GSR series of routers are two completely different architectures and normally fulfill different roles. The 7500 is considered an 'edge' class router, where it usually sits on the edge of an end-users network providing service less than OC-3. The 12000 GSR is a 'carrier-class' router that would commonly be found in the network of a service provider on OC-12 or greater links.

Next, the router needs to know what action to take when a certain class of traffic is to be queued for transmission. This is done with the *policy-map* command:

```
policy-map sdsu_out
 description SDSU QoS Output Policy
  class qbss
    bandwidth percent 1
    queue-limit 32
  class Premium
    priority percent 30
  class class-default
    random-detect
!
```

This *policy-map* command instructs the router what to do when a packet is to be queued for transmission according to the defined classes. The above command represents a typical configuration for all three South Dakota universities and the DS-3 to the GPN router in Minneapolis.

Any traffic matching the QBSS class is to receive a minimum of one percent of the bandwidth available *should congestion occur* outgoing on an interface. It is important to note that IOS does not perform any queuing functions unless the interface is experiencing congestion, expect with priority queuing. This is an assumption that there is no need to enforce limits unless there is other traffic of higher priority.

The Premium policy definition gives traffic marked by class-map *Premium* low latency queuing where a maximum of 30 percent of the available bandwidth if severe congestion occurs. This ensures that the priority queue servicing will not starve other queues. This feature will force the router to always service the priority queue first, even if there is no congestion on an interface.

The last category defines how all other traffic is serviced, which is with a weighted Random Early Detect algorithm. IP Precedence values are taken into consideration with weighted RED, and the higher the priority, the less chance a packet will get dropped should traffic levels reach congestion thresholds.

On the Cisco 7500 series router running IOS 12.1E, there is a command that displays queuing statistics according to policy-maps:

```
sd.r#sh policy-map interface atm 4/0/0.132

  service-policy output: common

    queue stats for all priority classes:
      queue size 0, queue limit 138
      packets output 495061, packet drops 244
      tail/random drops 0, no buffer drops 0, other drops 244

    class-map: qbss (match-any)
      46492886 packets, 5457746595 bytes
      5 minute offered rate 161000 bps, drop rate 0 bps
      match: ip dscp 8
        46492887 packets, 5457748107 bytes
```

```
    5 minute rate 161000 bps
  match: access-group name news
    0 packets, 0 bytes
    5 minute rate 0 bps
  queue size 0, queue limit 32
  packets output 46490990, packet drops 1894
  tail/random drops 1894, no buffer drops 0, other drops 0
  bandwidth: 1%, kbps 320
  queue-limit 32
  random-detect:
    Exp-weight-constant: 9 (1/512)
    Mean queue depth: 0
    Class Random       Tail   Minimum   Maximum     Mark       Output
            drop       drop threshold threshold  probability  packets
    0          0          0         1         2     1/10            0
    1        271        467         1         2     1/10     46490990
    2          0          0         1         2     1/10            0
    3          0          0         1         2     1/10            0
    4          0          0         1         2     1/10            0
    5          0          0         1         2     1/10            0
    6          0          0         1         2     1/10            0
    7          0          0         1         2     1/10            0

  class-map: Premium (match-any)
    495305 packets, 697460835 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    match: ip dscp 46
      492133 packets, 696938090 bytes
5 minute rate 0 bps
    match: ip precedence 5
      3166 packets, 522337 bytes
      5 minute rate 0 bps
    match: access-group name icmp
      6 packets, 408 bytes
      5 minute rate 0 bps
    Priority: 30% (9600 kbps), burst bytes 240000, b/w exceed drops: 244

  class-map: class-default (match-any)
    443392758 packets, 170580697406 bytes
    5 minute offered rate 181000 bps, drop rate 0 bps
    match: any
      443392757 packets, 170580695902 bytes
      5 minute rate 181000 bps
    queue size 0, queue limit 319
    packets output 443385082, packet drops 19315
    tail/random drops 19313, no buffer drops 0, other drops 2
    random-detect:
      Exp-weight-constant: 9 (1/512)
      Mean queue depth: 0
      Class Random       Tail   Minimum   Maximum     Mark       Output
              drop       drop threshold threshold  probability  packets
      0       1079        466        79       159     1/10     439306791
      1         20          0        88       159     1/10         99940
      2          0          0        98       159     1/10          9508
      3       3146      14514       108       159     1/10        181562
      4          0          0       118       159     1/10         59851
      5          0          0       128       159     1/10             0
      6          0         19       138       159     1/10       3722787
      7          0          0       148       159     1/10          2703
```

This information is also available via Cisco's SNMP private enterprises MIB *CLASS-BASED-QOS-MIB.my*. With this information available via SNMP, it is possible to capture and graph, as seen in Figure 10.

To verify that QoS is working as defined, several tests were performed, and the results graphed in Figure 10. Essentially three flows of traffic were initiated between several hosts, and the output was taken from the flow-generating tool called IPerf. [IPERF]

The first flow initiated was considered the long term, less-than-best-effort flow. This flow would simulate a scientist attempting to move terra-bytes of information across a network without impacting the performance of the network for other users. The IP traffic generated by this flow is marked with DSCP 8. A second flow is then started and is considered best effort. This flow would represent a short-term FTP session lasting less than a couple of minutes.

Last, a high-priority stream is generated and marked with DSCP 46. This would simulate a videoconference or voice over IP session in which the total amount of traffic is not allowed to exceed 30% of the available bandwidth. TCP was used to generate the high-priority flow and probably does not accurately represent a videoconference or VoIP session, which is normally UDP. Instead, TCP was selected because it was important to see how the flow would be allowed to expand within the amount of allocated bandwidth. In general, the router acted as expected and treated the traffic appropriately.

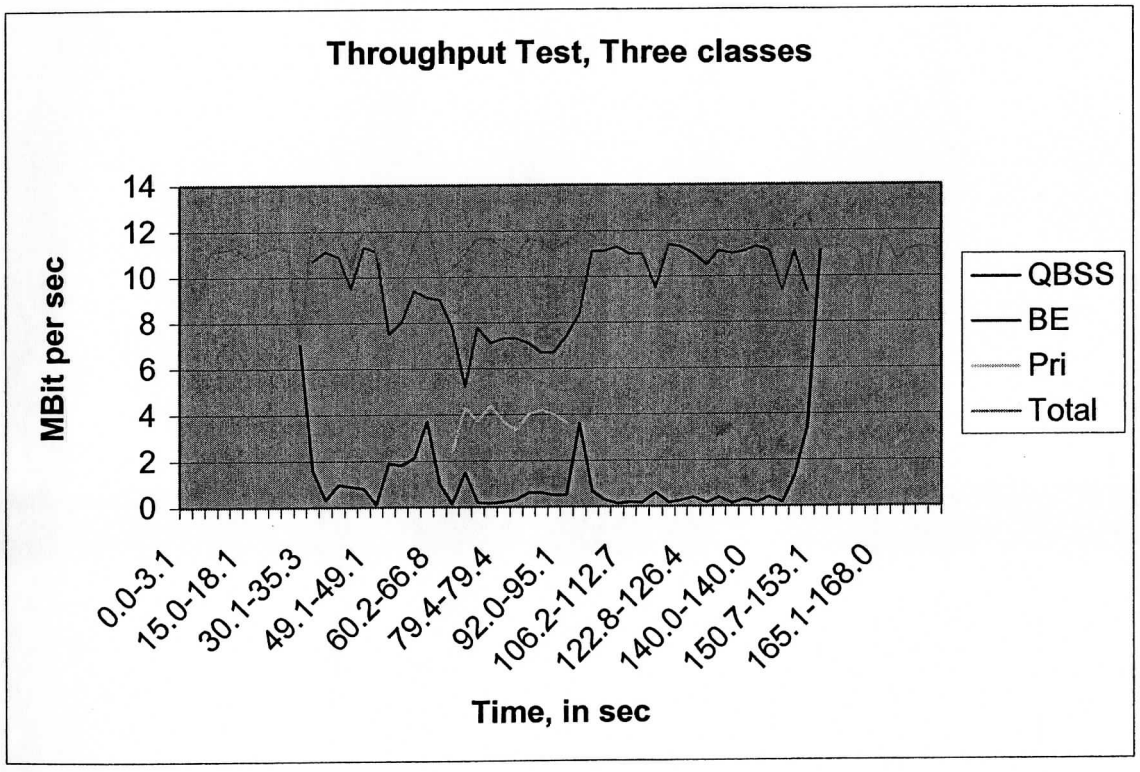


Figure 10- Throughput Testing with three different classes of traffic defined.

Traffic Visualization

It is often required that usage statistics of a router interface be counted or displayed to help the administrator of the router troubleshoot and plan for provisioning. Two methods for monitoring the queuing activity on the Cisco 7500 series router were immediately available. The first tool, Cisco QoS Device Manager 2.0 (QDM), and the second tool was a custom implementation of RRDTool in conjunction with remote SNMP monitoring.

Cisco QDM 2.0 is an additional software package that is installed and supported on various Cisco router platforms, including the Cisco 7500 series router. It presents a graphical interface to the user via HTTP and Java at a remote monitoring station, and presents near-real-time data in the form of graphs to the user.

QDM allows for near real-time monitoring of the activity on an interface. Graphs can be generated from interface statistics such as bits transferred per second, bytes transferred per second, bit and packet counts, queue statistics such as drops, overruns, and depth, and RED statistics. Multiple graphs can be generated and displayed in near real-time so the QDM user can visualize what activity is occurring on an interface. Data can be manually exported into spreadsheet format for later viewing.

QDM has the ability to independently graph multiple values and display them in time-synchronized windows to the user. Figure 11 represents an actual display of QDM graphing best effort, less-than-best effort, and priority traffic.

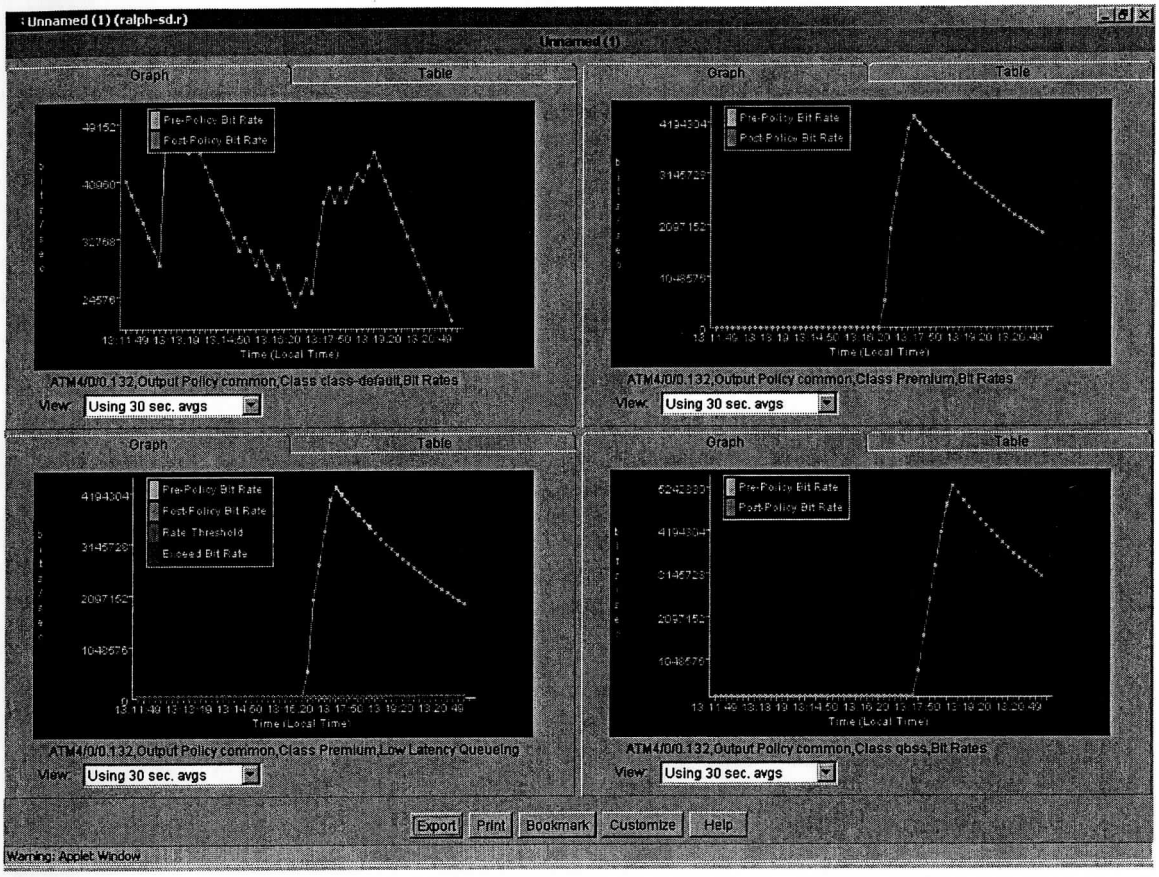


Figure 11- QDM monitoring BE, LBE and Pri Traffic in near real-time.

Cisco's QDM 2.0 has the ability to monitor in near real-time the activity of several important statistics, but it does not function adequately for monitoring and archiving activity over a long periods of time. In order to accomplish such goals as long-term archiving, RRDTool was used with SNMPTools to archive and graph QoS activities on the SDREN router. This method is quite similar to the CAR monitoring used on the Great Plains Network GSR mentioned in Section 7 of this paper.

The Cisco MIB, *CLASS-BASED-QOS*, was used by a remote monitoring station to remotely poll the router for values of interest. The OID of the most relevant table entry in this project is:

```
"cbQosCMPrePolicyByte"        "1.3.6.1.4.1.9.9.166.1.15.1.1.5"
```

There were three instances of this table, one for each class-of-service on an interface. For instance, SDSU's complete OID for the premium, default, and less-than-best effort classes of service are:

```
Priority: 1.3.6.1.4.1.9.9.166.1.15.1.1.5.1827403.1827420
Default: 1.3.6.1.4.1.9.9.166.1.15.1.1.5.1827403.1827404
QBSS: 1.3.6.1.4.1.9.9.166.1.15.1.1.5.1827403.1827412
```

These MIB entities are polled every minute and their values are placed in a Round Robin Database, RRDTool. RRDTool also provides the graphing function every minute, so the graphs generated are accurate up to the minute. This allows network administrators to see a fairly accurate representation of what QoS activity is occurring on the Cisco 7500 router that provides Internet 2 access to SDREN. Figure 12 is an actual graph generated by using this method.

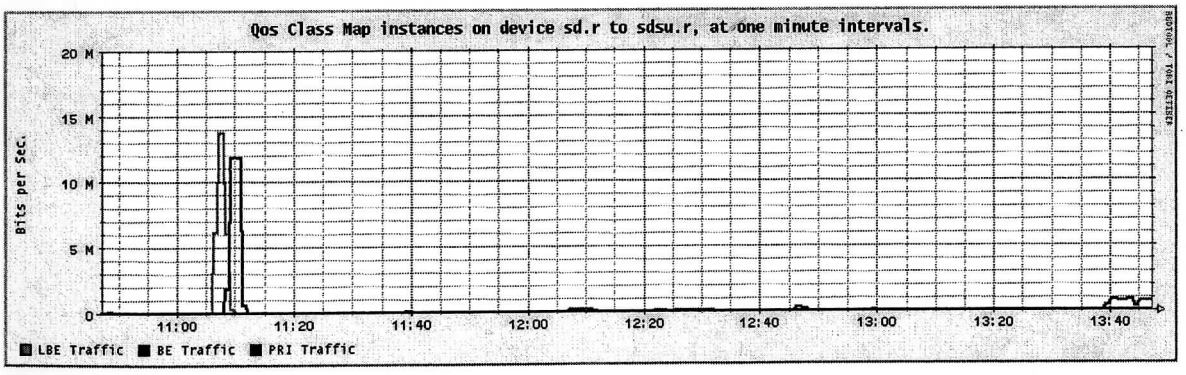


Figure 12- Graph representing the different classes of service

Conclusions

Traffic Engineering methods are available today that can increase the performance of IP networks and the Internet. Two methods of Traffic Engineering are Multiprotocol Label Switching and Quality of Service. Emerging applications on the Internet today suffer from the lack of sufficient resources such as bandwidth, latency, and packet loss.

A multi-service Internet is a network that is capable of carrying voice, video and data traffic, all at the same time, with high-quality experiences perceived by the users. Several architectures have been developed to accomplish a multi-service network, and it may be several years before the results are finally realized.

Several reasons exist why these new applications are emerging. One reason is that the technology to drive these applications is readily available. New advances in computing power, algorithm development, and commodity computing prices make these tools available at low cost with high-value.

A second reason why these new applications are becoming abundant is that they are enabling people to do new things more easily and cost effectively. Using email instead of postal courier saves time. Commerce web sites reach new customers and reach current customers more quickly and more cost effectively. It is now possible to make a phone call over the Internet for free. It is also possible to have a videoconference with someone around the world at very low cost.

A multi-service network could be accomplished by providing an infinite amount of bandwidth to the users. While this may provide a network infrastructure robust enough to handle a plethora of services like voice, video and data, it is unclear whether such a network can be deployed. Currently there is a 'fiber glut', with the telecommunications carriers over-provisioning fiber optic cable for the potential future needs. But, there is a lack of demand for these resources and there is a lack of switching and routing technology to light the fiber.

It would be unwise to assume that an infinite amount of bandwidth will be available. Even if a substantial amount of over provisioning is available in the future, it is still unclear whether or not this would provide all of the guarantees needed to run a multi-service network.

At the edge of the Internet, where the real usage occurs, there are insufficient network resources available to those who need it the most: businesses, universities, governments, and individual users. While there is not necessarily a shortage of network services at the edge, there is limited availability, and the availability is cost based.

The current model of the Internet is considered "best-effort". The network will make every attempt to deliver data in the form of packets to the end destination, and it does not guarantee that the data packet will be received or when. It may be argued that the best-

effort model of the Internet substantially contributed to its success, but that model may be dated with the arrival of new applications like video conferencing.

Multiprotocol Label Switching can be used in a network were traffic loads are unevenly distributed. MPLS, as a traffic-engineering tool, can be used to 'detour' traffic across under-utilized network links, balancing the load of traffic and increasing the perceived quality of service to the network's users.

While MPLS may solve certain problems, it is a deviation from the standard IP destination based routing model, and should only be used when other methods of traffic engineering have been exhausted. MPLS alters the IP routing model, which may disrupt the natural flow of the Internet. MPLS may scale well within a single provider's network, but there is a lack of scalable network-to-network protocols for exchanging MPLS path information, and MPLS is not an end-to-end solution.

Quality of Service algorithms in the form of packet queuing provides an alternative method to increase the quality of a network for the users. With the use of alternative queuing mechanisms in the network, priority levels can be altered and flows of traffic can be treated differently depending on attributes like IP source and destination, protocol, and type of service.

This project was able to successfully deploy, test and measure several QoS functions like priority queuing and less-than-best effort queuing on the South Dakota Research and Education Network. The queuing algorithms provided by Cisco Systems worked well in the high-speed ATM WAN environment on the 7500 series IP router. Several methods of measurement and recording were devised and implemented, and measurement is considered to be an ongoing goal of this project. Information is being provided to the universities and organizations whose services were used in this project.

At the edges the Great Plains Network, alternate queuing functions have been deployed on the Packet Over SONET OC-12 interfaces to Abilene in Kansas City and Minneapolis. This alternate queuing provides strict, low latency queuing and an additional class of less-than-best effort queuing for the outgoing traffic. Measurement tools are available but limited, and provide only rudimentary information due to the architecture of the Cisco Gigabit Switch Router.

Traffic Engineering can be used to increase the performance of existing networks, usually with existing network hardware and software. QoS and MPLS can be leveraged instead of procuring additional network infrastructure, resulting in a substantial cost savings. Traffic Engineering adds value to the network for the users, and provides leverage to network providers with obtaining new customers.

It is unknown if the current QoS technologies will scale to the entire Internet. In order to achieve high-quality end-to-end performance, implementation of QoS queuing mechanisms must be completed. New protocols could be developed that aid in the deployment of QoS

on a grand scale, and QoS will continue to be the focus of next-generation networks well into the 21st century.

References:

[AVVID] *Cisco AVVID White paper:* http://www.cisco.com/warp/public/cc/so/neso/vvda/iptl/avvid_wp.htm

[TEWG1] Definition of Traffic Engineering, www.ietf.org

[Huston00] *Internet Performance Survival Guide*, Geoff Huston, Wiley, 2000, pp. 6-17

[CiscoGTS] *Cisco Policing and Shaping Overview* http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart4/qcpolts.htm#15047

[FloydVanRED93] *Random Early Detection (RED) gateways for Congestion Avoidance,* S. Floyd and V. Jacobson, 1993. http://www.aciri.org/floyd/papers/red/red.html

[SMD98] Effects of Random Early Detection on Router Interfaces, http://adm.ebone.net/~smd/red-1.html

[AbQoS1] *Draft Abilene Premium Service Test program,* http://www.internet2.edu/abilene/qos/aps.pdf

[QBSSdfn] *QBone Scavenger Service (QBSS) Definition,* QBSS Working Group, March 2000, http://qbone.internet2.edu/qbss/qbss-definition.txt

[IPERF] *The TCP/UDP Bandwidth Measurement Tool*, National Laboratories for Advanced Network Research DAST Project, http://dast.nlanr.net

[whatis1] *Definition of Real-Time*, http://whatis.techtarget.com/definition/0,289893,sid9_gci214344,00.html

[floydECN] *The Addition of Explicit Congestion Notification (ECN) to IP*, S. Floyd, Proposed IETF Draft Standard. http://www.aciri.org/floyd/papers/draft-ietf-tsvwg-ecn-04.txt

[shcreedharDRR] *Efficient Fair Queuing Using Deficit Round Robin*,M. Shreedhar and G. Vargheese. http://ipoint.vlsi.uiuc.edu/wireless/papers-p/p231-shreedhar.pdf

Appendix A: Monitoring Code

What follows in this appendix are the scripts written to graph and store SNMP data of QoS network activity. RRDTool was used in conjunction with UCD SNMP Tools. Perl was used to call the system utilities, process the text into data, manipulate the data, insert the data into RRDTool via systems calls, and generate the graphs with a second RRDTool command.

An upcoming version 2 of these scripts will most likely use Perl's SNMP Module, so that a system call to the UCD SNMP Tools would not be required. Formatting and parsing of the text would then not be required, perhaps increasing performance and reducing processing time.

RRDTool now has a Perl Module available that allows RRDTool functions to added to Perl as function calls. This could also reduce processing time, and simplify the code.

```
# The following script polls and stores Cisco QoS Class Map
# information from a Cisco router using the CLASS-BASED-QOS-MIB.my
# MIB.
#
# Written by Dave Hartzell
# May 2001
# Version 1.0
#
#!/usr/bin/perl -w

# Do a system call and run SNMPGET
open (BE, "snmpget 192.41.204.1 dakota
enterprises.9.9.166.1.15.1.1.5.1827277.1827294|");
open (LBE, "snmpget 192.41.204.1 dakota
enterprises.9.9.166.1.15.1.1.5.1827277.1827278|");
open (PRI, "snmpget 192.41.204.1 dakota
enterprises.9.9.166.1.15.1.1.5.1827277.1827286|");

# Put the output into a variable…
$BEout = (<BE>);
$LBEout = (<LBE>);
$PRIout = (<PRI>);

# Strip (out of the variable) the counter value
if ($BEout =~ /(\d+)\s\=\s(\d+)/) {$BEout = $2}
if ($LBEout =~ /(\d+)\s\=\s(\d+)/) {$LBEout = $2}
if ($PRIout =~ /(\d+)\s\=\s(\d+)/) {$PRIout = $2}

# Store the counter value into a Round Robin Database
# by doing a system call…
system "/home/hartzell/rrdtool-1.0.33/bin/rrdtool update
/home/hartzell/snmp/i2-out-pol.rrd N:$LBEout:$PRIout:$BEout";

# Now graph the data with RRDTool by doing a system call
system "/home/hartzell/rrdtool-1.0.33/bin/rrdtool graph -a GIF
/home/hartzell/snmp/i2-qos.gif --start -3600 \\
```

```
    -w 800 -h 200 --title=\"Qos Class Map instances on device  sd.r to
    mn-2.r, at one minute intervals.\" \\
    -v\"Bits per Sec.\" \\
    DEF:lbe=/home/hartzell/snmp/i2-out-pol.rrd:lbe:AVERAGE \\
    DEF:be=/home/hartzell/snmp/i2-out-pol.rrd:be:AVERAGE \\
    DEF:pri=/home/hartzell/snmp/i2-out-pol.rrd:pri:AVERAGE \\
    CDEF:lbebits=lbe,8,* \\
    CDEF:bebits=be,8,* \\
    CDEF:pribits=pri,8,* \\
    LINE2:lbebits#00FF00:\"LBE Traffic\" \\
    LINE2:bebits#FF0000:\"BE Traffic\" \\
    LINE2:pribits#0000FF:\"PRI Traffic\" ";

close (BE);
close (LBE);
close (PRI);

#
# The following command (run through a UNIX Shell) creates the database
# that was used with the Cisco Class Map QoS data.  The command defines
# three counters or Data Sources, lbe, pri, and be, and keeps them around
# for approximately 73 days.  This command is run only once to create the
# database/archive.
#
# Graphs are usually only generated with the last couple hours on the plot,
# but with 70+ days in the archive, it is possible to look back over time
# for events or trends in the data.
#
# It would be nice if the author would write a web/CGI interface so the
# users of the data could specify when and what data they would like to see,
# and have a custom graph generated...
#
/home/hartzell/rrdtool-1.0.33/src/rrdtool create i2-out-pol.rrd --step 60\
 DS:lbe:COUNTER:70:U:U \
 DS:pri:COUNTER:70:U:U \
 DS:be:COUNTER:70:U:U \
 RRA:AVERAGE:0.5:1:105120 \
 RRA:MAX:0.5:1:105120
```

```
# The following script polls and stores Cisco CAR information
# from a Cisco router using the CISCO-CAR-MIB.my MIB file.
#
# Written by Dave Hartzell
# June 2001
# Version 1.0
#!/usr/bin/perl -w

open (in46sw,  "snmpget ks-2.r.greatplains.net 1g2p3
enterprises.9.9.113.1.2.1.1.2.2.1.1|");
open (in8sw,   "snmpget ks-2.r.greatplains.net 1g2p3
enterprises.9.9.113.1.2.1.1.2.2.1.2|");
open (out46sw, "snmpget ks-2.r.greatplains.net 1g2p3
enterprises.9.9.113.1.2.1.1.2.2.2.1|");
open (out8sw,  "snmpget ks-2.r.greatplains.net 1g2p3
enterprises.9.9.113.1.2.1.1.2.2.2.2|");

open (in46fl,  "snmpget ks-2.r.greatplains.net 1g2p3
enterprises.9.9.113.1.2.1.1.4.2.1.1|");
```

```
open (in8fl,   "snmpget ks-2.r.greatplains.net 1g2p3
enterprises.9.9.113.1.2.1.1.4.2.1.2|");
open (out46fl, "snmpget ks-2.r.greatplains.net 1g2p3
enterprises.9.9.113.1.2.1.1.4.2.2.1|");
open (out8fl,  "snmpget ks-2.r.greatplains.net 1g2p3
enterprises.9.9.113.1.2.1.1.4.2.2.2|");

$in46sw = (<in46sw>);
$in8sw = (<in8sw>);
$out46sw = (<out46sw>);
$out8sw = (<out8sw>);

$in46fl = (<in46fl>);
$in8fl = (<in8fl>);
$out46fl = (<out46fl>);
$out8fl = (<out8fl>);

if ($in46fl =~ /(\d+)\s\=\s(\d+)/) {$in46fl = $2}
if ($in8fl =~ /(\d+)\s\=\s(\d+)/) {$in8fl = $2}
if ($out46fl =~ /(\d+)\s\=\s(\d+)/) {$out46fl = $2}
if ($out8fl =~ /(\d+)\s\=\s(\d+)/) {$out8fl = $2}

if ($in46sw =~ /(\d+)\s\=\s(\d+)/) {$in46sw = $2}
if ($in8sw =~ /(\d+)\s\=\s(\d+)/) {$in8sw = $2}
if ($out46sw =~ /(\d+)\s\=\s(\d+)/) {$out46sw = $2}
if ($out8sw =~ /(\d+)\s\=\s(\d+)/) {$out8sw = $2}

# Here we need to do some math, since we have a conformed counter and an
# exceeded variable.  To get the true representation of what is going on,
# we need to add them up.

$in46 = $in46fl + $in46sw ;
$in8 = $in8fl + $in8sw;
$out46 = $out46fl + $out46sw;
$out8 = $out8fl + $out8sw;
system "/home/hartzell/rrdtool-1.0.33/bin/rrdtool update /home/hartzell/snmp/ks2-
car-pol.rrd N:$in8:$out8:$in46:$out46";

system "/home/hartzell/rrdtool-1.0.33/bin/rrdtool graph -a GIF
/home/hartzell/snmp/ks-2-car.gif --start -108
00 \\
    -w 800 -h 200 --title=\"CAR DSCP instances on device KS-2.r pos 0 0, at one
    minute intervals.\" \\
    -v\"Bits per Sec.\" \\
    DEF:in8=/home/hartzell/snmp/ks2-car-pol.rrd:in8:MAX \\
    DEF:out8=/home/hartzell/snmp/ks2-car-pol.rrd:out8:MAX \\
    DEF:in46=/home/hartzell/snmp/ks2-car-pol.rrd:in46:MAX \\
    DEF:out46=/home/hartzell/snmp/ks2-car-pol.rrd:out46:MAX \\
    CDEF:in8bits=in8,8,* \\
    CDEF:out8bits=out8,8,* \\
    CDEF:in46bits=in46,8,* \\
    CDEF:out46bits=out46,8,* \\
    LINE2:in8bits#00FF00:\"Incoming DSCP 8 Traffic\" \\
    LINE2:out8bits#FF0000:\"Outgoing DSCP 8 Traffic\" \\
    LINE2:in46bits#0000FF:\"Incoming DSCP 46 Traffic\" \\
    LINE2:out46bits#FF00FF:\"Outgoing DSCP 46 Traffic\" ";

close (in8fl);
close (out8fl);
close (in46fl);
close (out46fl);
close (in8sw);
close (out8sw);
```

```
close (in46sw);
close (out46sw);


# The following shell will create the RRD Database/Archive for
# a Cisco GSR having 4 CAR variables.  In this case, we would like
# to use two Input CAR counters and two Output CAR counters.
#
/home/hartzell/rrdtool-1.0.33/src/rrdtool create mn2-car-pol.rrd --step 60\
 DS:in8:COUNTER:70:U:U \
 DS:out8:COUNTER:70:U:U \
 DS:in46:COUNTER:70:U:U \
 DS:out46:COUNTER:70:U:U \
 RRA:AVERAGE:0.5:1:105120 \
 RRA:MAX:0.5:1:105120
```

Appendix B: Project Timeline

This appendix represents a time-line for the implementation of Quality of Service on the Great Plains Network and the South Dakota Research and Education Network. Each block represents one week, and the X marks a milestone.